Coding Challenge

General

This task is designed to assess skill and experience as a developer and your ability to think critically.

- You will be evaluated both on how it functions and how the code is written.
- You may use any language, framework, platform, or tool you like.
- You **will not** be evaluated on the aesthetics of the project, provided they do not interfere with other aspects of the project.
- You should adhere to all general UI standards.
- You should expect to spend an hour on this challenge.
- You may spend as much time as you like, but having more complete does not mean a better score.
- You are **not** required to complete tasks in order and may skip any task and complete ones after it if you so chose.
- You are **not** required to complete all aspects of the assessment.
- You may use any resources you like, **including other people**, provided you do not copy other works, can explain the resulting code, and can defend the decisions you made.
- No matter how much or how little you have done, **ensure it runs**.

Tasks

1. Parse the provided JSON file containing information about the states.
2. Create a list of all states. The layout should be easily modified, but initially each item in the list should contain:
   A. The state's name.
   B. An image of the state's flag, loaded from the provided url. **Be sure to preserve the original image aspect ratio.**
3. When the user selects an item from the list, present a new screen with more detailed information about that state. That screen must also have a way to navigate back to the original list, such as a back button.
4. On the detail screen, add a toggle to allow the user to indicate if they have visited that state or not.
5. On the detail screen, add the ability to select a date (month, day, and year) when the user first visited that state. A user may not enter a date on states they have not visited and if a user unmarks the state, the date must also be cleared.
6. Add an indicator to the list screen that shows if the state has been visited. It must update appropriately when the user toggles a state on the detail screen.
7. Add a button to clear all visited states.
8. Ensure the list of visited states persists between launches.
9. Add sorting and filtering options to the list of states. Suggestions might be to sort by visited states first, alphabetical, or most populous states first.